OUTPUT:

1. CPP Program to Print Fibonacci Number Series

#include <iostream>

using namespace std;

int main ()

{

int num = 7, i, current;

int last = 0, prev = 1;

cout << "Fibonacci Series : ";

for (i = 1; i <= num; i = i + 1) {

cout << last << "\t";

current = last + prev;

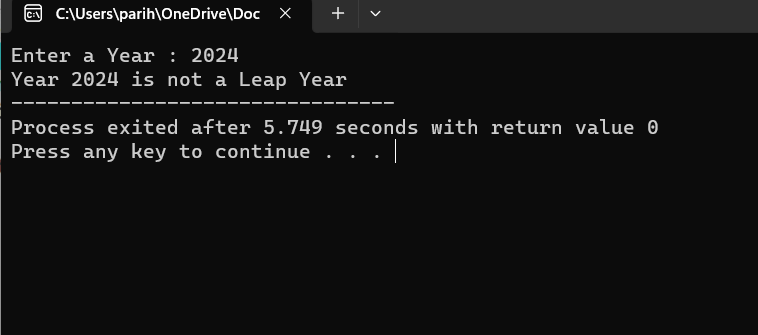
last = prev;

prev = current;

}

return 0;

}

OUTPUT:

1. CPP Program to check whether the year entered is Leap Year or not.

#include <iostream>

using namespace std;

int main () {

int year;

cout << "Enter a Year: ";

cin >> year;

if ((year % 400 == 0) && (year % 4 == 0 || year % 100 !== 0)) {

cout<<"Year "<< year << " is a Leap Year";

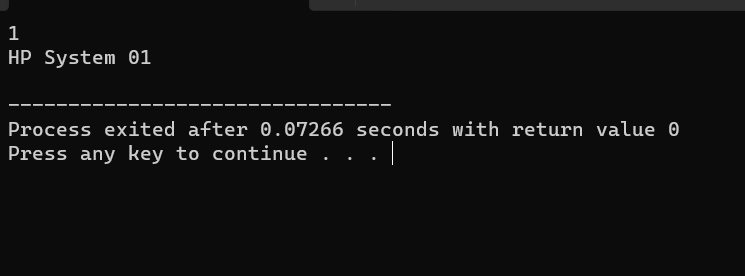
} else {

cout<<"Year "<< year <<" is not a Leap Year";

}

return 0;

}

OUTPUT:

1. CPP Program to demonstrate Classes and Objects.

#include <iostream>

using namespace std;

class System {

public:

int sys\_no;

string System\_name;

};

int main () {

System sys1;

sys1.sys\_no = 01;

sys1.System\_name = "HP System 01";

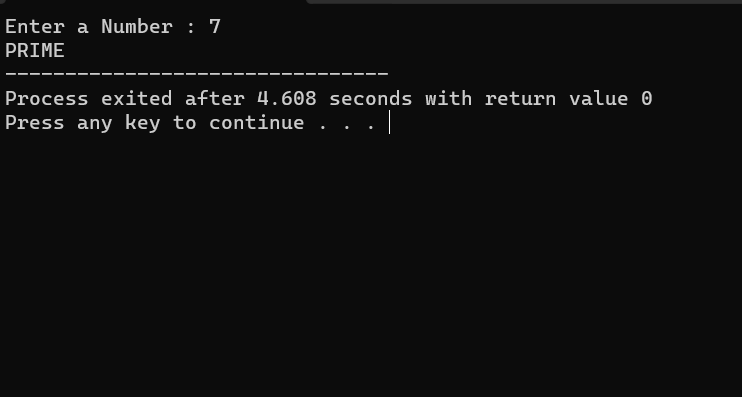
cout<<sys1.sys\_no<<endl;

cout<<sys1.System\_name<<endl;

return 0;

}

OUTPUT:



1. CPP Program to check entered number is prime or not.

#include <iostream>

using namespace std;

int main ()

{

int num;

cout << "Enter a Number: ";

// Taking input from the User

cin >> num;

if (num < 2)

{

cout << "NOT PRIME";

return 0;

}

else

{

for (int i = 2; i < num; i++)

{

if (num % i == 0) {

cout << "NOT PRIME";

return 0;

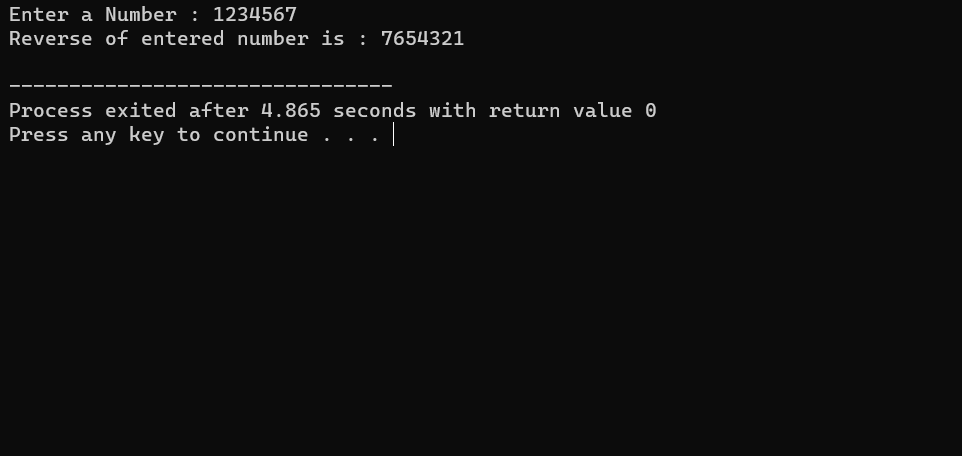
}

}

cout << "PRIME";

}

}

OUTPUT:

1. CPP Program to print the reversed number of entered number.

#include <iostream>

using namespace std;

int main () {

int num, ans = 0, rem;

cout << "Enter a Number: ";

// Taking input from User

cin >> num;

while (num > 0) {

rem = num % 10;

num /= 10;

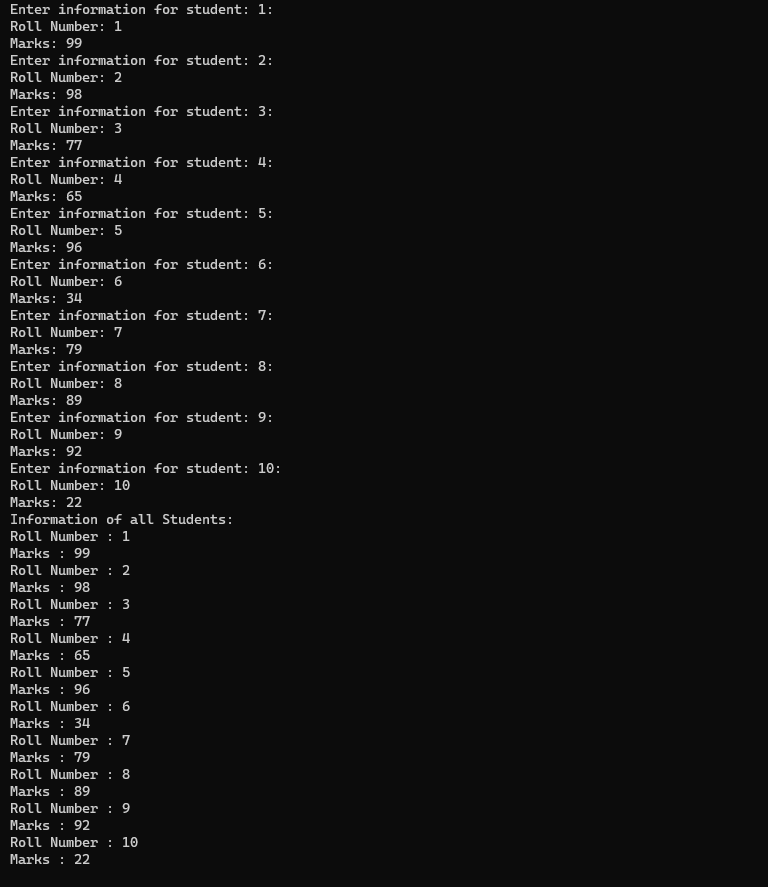
ans = ans \* 10 + rem;

}

cout << "Reverse of entered number is: " << ans << endl;

return 0;

}

OUTPUT:

1. CPP Program to store the information of 10 Students using Object arrays in CPP Classes.

#include <iostream>

using namespace std;

// Declaring a student class

class Student {

public:

int rollNumber;

float marks;

void setInfo (int rollNumber, float marks) {

this -> rollNumber = rollNumber;

this -> marks = marks;

}

void displayInfo () {

cout << "Roll Number: " << rollNumber << endl;

cout << "Marks: " << marks << endl;

}

};

int main (){

Student students [10];

for (int i = 0; i < 10; i = i + 1) {

int rollNumber;

float marks;

cout << "Enter information for student: "<< (i + 1) << ":" << endl;

cout << "Roll Number: ";

cin >> rollNumber;

cout << "Marks: ";

cin >> marks;

students[i]. setInfo (rollNumber, marks);

}

cout << "Information of all Students: "<<endl;

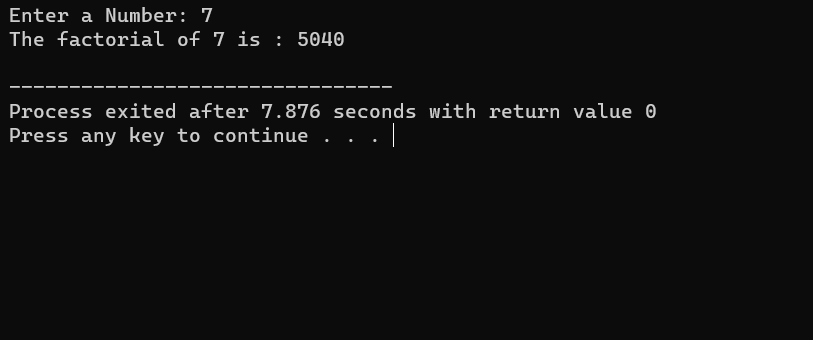
for (int i = 0; i < 10; i = i + 1) {

students[i]. displayInfo ();

}

return 0;

}

OUTPUT:

1. CPP Program to find the factorial of entered number.

#include <iostream>

using namespace std;

int main () {

int num, i, fact = 1;

cout << "Enter a Number: ";

// Taking input from User

cin >> num;

if (num == 0) {

cout << "The factorial of 0 is 1"<<endl;

}

else if (num < 0) {

cout << "Please Enter a Number that is greater than or equal to 0"<<endl;

} else {

for (i = 1; i <= num; i = i + 1) {

fact \*= i;

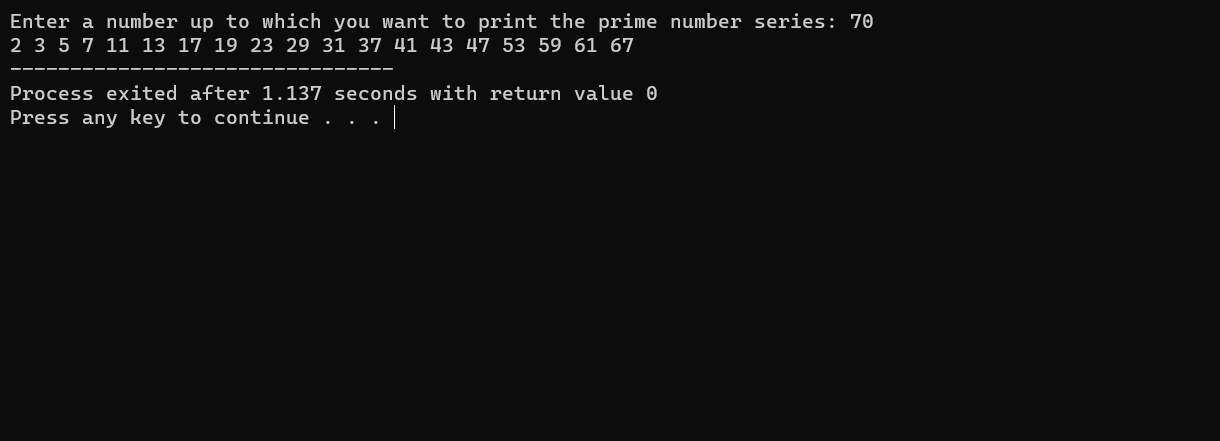
}

cout << "The factorial of "<< num << " is: "<< fact<<endl;

}

return 0;

}

OUTPUT:

1. CPP Program to print Series of Prime Numbers.

#include <iostream>

using namespace std;

bool isPrime (int number) {

if (number < 2) {

return false;

} else {

for (int i = 2; i \* i <= number; i++) {

if (number % i == 0) {

return false;

}

}

}

return true;

}

void displayPrimes (int number) {

for (int i = 2; i <= number; i++) {

if (isPrime(i)) {

cout << i << " ";

}

}

}

int main () {

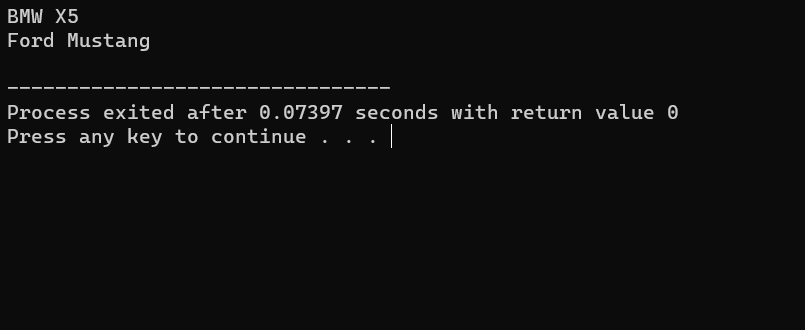
int num;

cout << "Enter a number up to which you want to print the prime number series: ";

cin >> num;

displayPrimes(num);

return 0;

}OUTPUT:

1. CPP Program to demonstrate constructors in CPP.

#include <iostream>

using namespace std;

class Car {

public:

string brand;

string model;

// Default constructor

Car (string brand, string model) {

cout << brand <<" " << model << endl;

}

};

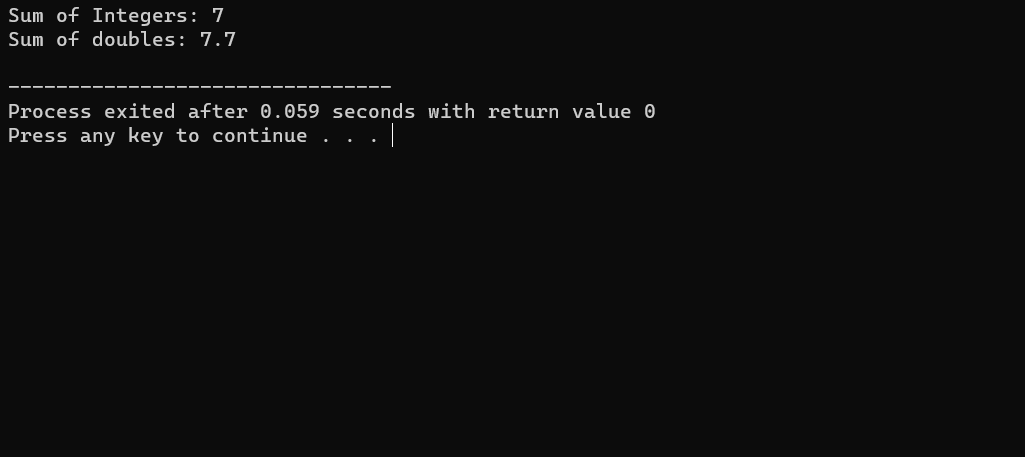
int main () {

Car One("BMW","X5");

Car Two ("Ford”, “Mustang");

return 0;

}

OUTPUT:

1. CPP Program to demonstrate Function Overloading

#include <iostream>

using namespace std;

int add (int a, int b) {

return a + b;

}

double add (double a, double b) {

return a + b;

}

int main () {

int sum1 = add (3,4);

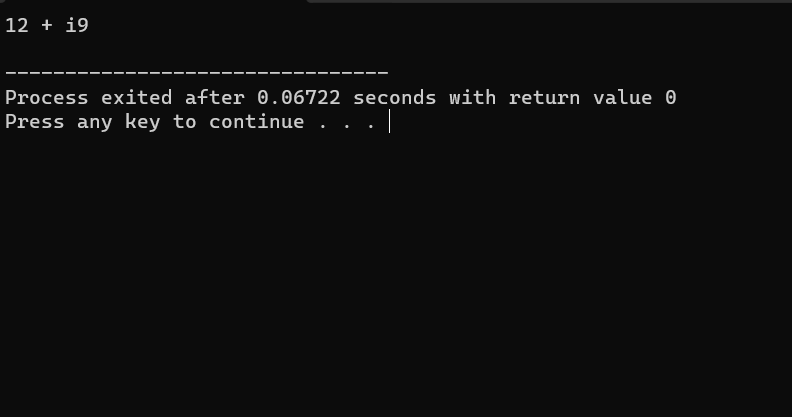
double sum2 = add (3.3,4.4);

cout << "Sum of Integers: " << sum1 << endl;

cout << "Sum of doubles: " << sum2 << endl;

return 0;

}

OUTPUT:

1. CPP Program to demonstrate Operator Overloading.

#include <iostream>

using namespace std;

class complexNum {

private:

int real, imag;

public:

complexNum (int r = 0, int i = 0) {

real = r;

imag = i;

}

complexNum operator+ (complexNum const& obj) {

complexNum res;

res. real = real + obj. real;

res. imag = imag + obj. imag;

return res;

}

void print () {

cout << real << " + i "<< imag << endl;

}

};

int main () {

complexNum c1(10, 5), c2(2, 4);

complexNum c3 = c1 + c2;

c3.print ();

return 0;

}